

WHAT IS CLAIMED IS:

1. A method which enables a user to prevent unauthorized access to files stored on a computer, comprising:

5 maintaining a first database which identifies files stored on the computer to be included in a safe zone;

maintaining a second database which defines authorized accesses to said files within said safe zone;

10 providing said computer with a filter;

upon a request for access to a file stored on said computer, utilizing said filter to access said first database and determine whether said file is within said safe zone; and

15 if said file is determined to be within said safe zone, accessing said second database to determine whether said request to access said file has been authorized.

2. A method as in claim 1, further comprising, if said request is determined to be unauthorized, then denying access to said file, else granting access to said file.

3. A method as in claim 2, further comprising, if access to said file is denied, then subsequently prompting said user to confirm or reverse said decision to deny access.

4. A method as in claim 3, wherein prompting said user to confirm or reverse said decision to deny access comprises indicating to said user an identity of an application that has requested access to said file.

5. A method as in claim 1, further comprising

providing an interface through which said user can update said first database.

6. A method as in claim 1, further comprising providing an interface through which said user can update said second database.

7. A method as in claim 1, further comprising encrypting said first database.

8. A method as in claim 1, further comprising encrypting said second database.

9. A method as in claim 1, wherein:

said first database is a distributed database, said distributed database comprising a file within each directory containing one or more of said files which were identified by said first database to be included within said safe zone; and

said filter accessing said first database comprises said filter accessing the files of said distributed database to verify whether said file for which access has been requested is within said safe zone.

10. A method as in claim 9, further comprising encrypting the files of said distributed database.

11. A method as in claim 1, further comprising, if said request for access is determined to have been made to a file within said safe zone, and if said request is determined to be authorized, then attempting to determine whether said request was initiated by a Trojan process.

12. A method as in claim 11, wherein attempting to determine whether said request was initiated by a Trojan

process comprises determining what application the request appears to be associated with, and also determining whether a timestamp which is associated with the request is consistent with one or more timestamps associated with the application's install.

13. A method as in claim 11, wherein attempting to determine whether said request was initiated by a Trojan process comprises determining whether a directory from which said request was launched is an appropriate location for the process making said request to be stored.

14. A method as in claim 1, wherein said filter is a part of an operating system which is installed on said computer.

15. A method as in claim 1, wherein said filter is only activated by remote queries to said computer.

16. Apparatus which enables a user to prevent unauthorized access to files stored on a computer, comprising:

- at least one computer readable storage media;
- and
- computer readable program code stored on said at least one computer readable storage media, said computer readable program code comprising:
 - program code for maintaining a first database which identifies files stored on said computer to be included in a safe zone;
 - program code for maintaining a second database which defines authorized accesses to said files within said safe zone;
 - program code for providing said computer with a filter;
 - program code for utilizing said filter to

20

access said first database and determine whether a file for which access has been requested is within said safe zone; and

program code for accessing said second database to determine whether said request to access said file has been authorized if said file is determined to be within said safe zone.

17. The apparatus of claim 16, further comprising program code for denying access to said file if said request is determined to be unauthorized, else for granting access to said file.

18. The apparatus of claim 17, further comprising program code for prompting said user, if access to said file denied, to confirm or reverse said decision to deny access.

5

19. The apparatus of claim 18, further comprising program code for indicating to said user an identity of an application that has requested access to said file when said user is prompted to confirm or reverse said decision to deny access.

20. The apparatus of claim 16, further comprising program code for creating a first interface through which said user can update said first database.

21. The apparatus of claim 16, further comprising program code for creating a second interface through which said user can update said second database.

22. The apparatus of claim 16, further comprising program code for encrypting said first database.

23. The apparatus of claim 16, further comprising

program code for encrypting said second database.

24. The apparatus of claim 16, further comprising program code for creating a distributed database comprising a file within each directory containing one or more of said files which were identified by said first database to be included in said safe zone, wherein said first database comprises said distributed database, and wherein said filter accessing said first database comprises said filter accessing the files of said distributed database to verify whether said file for which access has been requested is within said safe zone.

25. The apparatus of claim 24, further comprising program code for encrypting the files of said distributed database.

26. The apparatus of claim 16, further comprising program code for attempting to determine whether said request for access was initiated by a Trojan process if said request for access is determined to have been made to a file within said zone, and if said request for access is determined to be authorized.

27. The apparatus of claim 26, wherein the program code for attempting to determine whether said request was initiated by a Trojan process further comprises program code for determining what application said request appears to be associated with and for determining whether a timestamp which is associated with said request is consistent with one or more timestamps associated with the application's install.

28. The apparatus of claim 26, wherein the program code for attempting to determine whether said request was initiated by a Trojan process further comprises program

code for determining whether a directory from which said
5 request was launched is an appropriate location for the
process making said request to be stored.

29. The apparatus of claim 16, wherein said filter
is a part of an operating system which is installed on
said computer.

30. The apparatus of claim 16, wherein said filter
is only activated by remote queries to said computer.

31. An apparatus which enables a user to prevent
unauthorized access to files stored on a computer,
comprising:

5 means for identifying files stored on the
computer to be included in a safe zone;

means for defining authorized accesses to said
files within said safe zone;

10 means for determining whether a file for which
access has been requested is within said safe zone;
and

means for determining whether said request to
access said file has been authorized.